■ สุทธิศักดิ์ พงศ์ธนาพาณิช

SOFT MAIL

Advanced BASIC Programming Technique

: การทดสอบปัญหา Y2K ด้วย QuickBASIC

แนะนำวิธีการตรวจสอบปัญหา Y2K ด้วย QuickBASIC

บทความตอนนี้เป็นบทความที่ผู้เขียนต้องการจะแนะนำวิธีการตรวจสอบปัญหา Y2K (ปี ค.ศ. 2000) ที่กำลังเป็นที่สนใจในปัจจุบัน โดยนอกจากจะชี้ให้เห็นถึงที่มาของปัญหาแล้ว ในบทความตอนนี้ผู้เขียนยังได้ทำการเปิดเผยซอร์สโค้ดของโปรแกรม Y2KVIEW.EXE ซึ่งถูกเขียนด้วย QuickBASIC 4.50 ดังนั้นจึงนับได้ว่าบทความ Advanced BASIC Programming Technique ของนิตยสารไมโครคอมพิวเตอร์ เป็นบทความแรกของเมืองไทยที่ยอมเปิดเผยซอร์สโค้ดของโปรแกรมที่สามารถตรวจสอบปัญหา Y2K ของเครื่องพีซีทั่วไป และเพื่อความสะดวกของผู้เขียน ขอให้ผู้อ่านทุกท่านที่สนใจซอร์สโค้ดไม่ว่าจากบทความตอนใดก็ตาม กรุณาสั่งซื้อในรูปแบบของบริการซอฟต์เมล์ได้โดยตรง จากบริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน) เพราะผู้เขียนไม่สะดวกที่จะส่งซอร์สโค้ดเป็น e-mail ให้กับผู้อ่านแต่ละท่าน เนื่องจากต้องใช้เวลามากครับ

รู้จักกับ Real Time Clock

Real Time Clock (RTC) เป็นไมโครชิปนาฬิกาพร้อมปฏิทินที่ถูกใส่ลงในเครื่องคอมพิวเตอร์ระดับ AT ขึ้นไป ซึ่ง RTC นี้จะทำงานตลอดเวลา 24 ชั่วโมง โดยไม่สนใจว่าเครื่องคอมพิวเตอร์กำลังทำงานหรือถูกปิดอยู่ในขณะนั้น เนื่องจาก RTC จะมีแหล่งจ่ายไฟฟ้าเพื่อเลี้ยงวงจรให้ทำงานตลอดเวลาอยู่แล้ว และ

Advanced BASIC

ปฏิทินของ RTC ก็เป็นปฏิทินที่สามารถคำนวณหาปีอธิกสุรทิน (ที่มี 366 วัน) ได้อีกด้วย (ผู้อ่านสามารถศึกษาวิธีการคำนวณปฏิทินอย่างละเอียดได้จากหนังสือ QuickBASIC Toolbox ซึ่งแต่งโดยผู้เขียนเอง) และภายใน RTC จะมีหน่วยความจำประเภท CMOS RAM ขนาด 64 ไบต์ สำหรับจัดเก็บข้อมูลเกี่ยวกับนาฬิกาพร้อมปฏิทิน และรีจิสเตอร์ต่าง ๆ ดังมีรายละเอียดในตารางที่ 1

ตารางที่ 1
แสดงหน่วยความจำประเภท CMOS RAM ขนาด 64 ไบต์ สำหรับจัดเก็บข้อมูลเกี่ยวกับนาฬิกาพร้อมปฏิทิน

แอดเดรส (เลขฐานสิบหก)	รายละเอียด
00-0D	ข้อมูลนาฬิกาพร้อมปฏิทินของ RTC
0E	ไบต์สถานะ diagnostic
0F	ไบต์สถานะ Shutdown
10, 12	ไบต์ประเภทของฟลอปปี้ดิสก์ไดรฟ์ A, B และ C, D ตามลำดับ
11, 13	สงวนเอาไว้
14	ไบต์อุปกรณ์
15-16	ไบต์ขนาดหน่วยความจำหลัก
17-18	ไบต์ขนาดหน่วยความจำขยาย
19-2D	สงวนเอาไว้
2E-2F	ไบต์ CMOS checksum
30-31	ไบต์ขนาดหน่วยความจำขยาย
32	ไบต์ตัวเลข ค.ศ. 100 ปี (century) เช่น 19 สำหรับตั้งแต่ ค.ศ. 1900-1999 เป็นต้น
33	ไบต์แฟล็กข้อมูล (เมื่อมีการเปิดเครื่อง)
34	สงวนเอาไว้

ตารางที่ 2
แสดงข้อมูลที่ถูกจัดเก็บลงในไบต์ที่ 0 ถึง 13 (0DH) และ 50 (32H) ของ RTC

ไบต์ที่	รายละเอียด	แอดเดรส (เลขฐานสิบหก)
0	SECONDS	00
1	SEND ALARM	01
2	MINUTES	02
3	MINUTE ALARM	03
4	HOURS	04
5	HOUR ALARM	05
6	DAY OF WEEK (Sun-Sat)	06
7	DATE OF MONTH (1 - 31)	07
8	Month	08
9	Year (00-99)	09
10	Status Register A	0A
11	Status Register B	0B
12	Status Register C	0C
13	Status Register D	0D
50	Date Century (19)	32

จากตารางที่ 1 จะเห็นว่าในหน่วยความจำประเภท CMOS RAM ของ RTC จะมีเฉพาะไบต์ที่ 0 ถึง 13 (0DH) และ 50 (32H) เท่านั้น ที่เกี่ยวข้องกับนาฬิกาพร้อมปฏิทิน ดังนั้นในบทความตอนนี้ผู้เขียนจะอธิบายเฉพาะไบต์ดังกล่าวเท่านั้น โดยที่ข้อมูลที่ถูกจัดเก็บลงในไบต์ที่ 0 ถึง 13 (0DH) และ 50 (32H) ดังในตารางที่ 2

จากตารางที่ 2 จะเห็นว่า ถ้าหากเราต้องการอ่านค่านาฬิกาจาก RTC ก็จะอ่านค่าตัวเลขจากไบต์ที่ 0, 2 และ 4 ซึ่งจะได้ค่าตัวเลขของวินาที (0-59) นาที (0-59) และชั่วโมง (0-23) ตามลำดับ ส่วนการอ่านปฏิทินจาก RTC ก็จะอ่านค่าตัวเลขจากไบต์ที่ 6 ถึง 9 และ 50 ซึ่งจะได้ค่าตัวเลขของลำดับวันของสัปดาห์ตั้งแต่อาทิตย์ถึงเสาร์ (0-6) ลำดับวันที่ของเดือน (1-31) เดือน (1-12) ปี ค.ศ. (00-99) และตัวเลขศตวรรษ (date century) ซึ่งค่าตัวเลขนี้ได้ถูกกำหนดให้มีค่าเท่ากับ 19 โดยอัตโนมัติอยู่แล้ว และตัวเลขศตวรรษนี้เองเป็นที่มาของปัญหา Y2K ในปัจจุบัน เพราะในอีก 1 ปีข้างหน้า เราก็จะก้าวเข้าสู่ปี ค.ศ. 2000 ดังนั้นถ้าหากในวันที่ 1 มกราคม 2000 ตัวเลขศตวรรษของ RTC ยังคงมีค่าเท่ากับ 19 ก็แสดงว่าในวันดังกล่าวเรากำลังถอยหลังไปอยู่ ณ วันที่ 1 มกราคม 1900 นั่นเอง ด้วยเหตุผลเหล่านี้ ดังนั้นในการเขียนโปรแกรมเพื่อตรวจสอบปัญหา Y2K เราก็ต้องทำการตรวจสอบว่าข้อมูลนาฬิกาและปฏิทินของ RTC เกิดปัญหาดังกล่าวนี้หรือไม่ (Year 2000 Rollover problem) ถ้าหากเกิดปัญหาก็แสดงว่าเครื่อง PC ของผู้อ่านไม่ผ่านปี ค.ศ. 2000 ครับ แต่ถ้าหากเครื่อง PC ของผู้อ่านได้รับการแก้ไข เช่น โดยใช้ชิปแก้ไขปัญหา Y2K ที่ผลิตโดยคนไทยและมีจำหน่ายทั่วไป ก็จะสามารถผ่านปัญหานี้ไปได้

ระบบปฏิบัติการดอสและนาฬิกาของระบบ

เมื่อผู้อ่านทราบถึงแหล่งกำเนิดนาฬิกาและปฏิทินของเครื่อง PC แล้ว ต่อไปผู้เขียนจะอธิบายถึงนาฬิกาอื่น ๆ ของระบบปฏิบัติการดอส เพราะว่าภายใต้ระบบปฏิบัติการดอสจะมีนาฬิกาอยู่ทั้งหมด 3 ชุดด้วยกัน คือ RTC, BIOS Clock และ DOS Clock (หรือ System Clock) สำหรับ RTC ผู้เขียนได้กล่าวรายละเอียดไปแล้วในส่วนแรกของบทความ สำหรับในส่วนนี้ผู้เขียนก็จะขอกล่าวถึงเฉพาะนาฬิกาชุดที่เหลือ และความสัมพันธ์ระหว่างนาฬิกาทั้งสามชุด

สำหรับนาฬิกาของระบบปฏิบัติการดอส สามารถเรียกอีกอย่างได้ว่าเป็นนาฬิกาซอฟต์แวร์ เพราะได้ถูกคำนวณเวลาและวันที่โดยทางซอฟต์แวร์ โดยในการ

Advanced BASIC

คำนวณ ดอสได้มีการกำหนดข้อจำกัดเอาไว้ดังนี้ เนื่องจากนาฬิกา RTC สามารถมีค่า ค.ศ. ได้ตั้งแต่ 1900 ถึง 1999 แต่ระบบปฏิบัติการดอสได้ถูกนำมาใช้กับเครื่อง PC เป็นครั้งแรกเมื่อวันที่ 4 มกราคม 1980

ตารางที่ 3
แสดงการเข้าถึงนาฬิกาของระบบปฏิบัติการดอส ผ่านทางอินเทอร์รัพต์ 21H

ชื่อฟังก์ชัน	หมายเลขฟังก์ชัน	ค่าที่คืนกลับจากอินเทอร์รัพต์ 21H
การอ่านปฏิทิน RTC	AH = 2AH	AL = วันที่ของสัปดาห์ (0=Sun ,..., 6=Sat) CX = ปี ค.ศ. (1980 ถึง 2099) DH= เดือน (1 ถึง 12) DL = วันที่ (1 ถึง 31)
การตั้งปฏิทิน RTC	AH = 2BH CX = ปี ค.ศ. (1980 ถึง 2099) DH= เดือน(1 ถึง 12) DL = วันที่ (1 ถึง 31)	AL = 00H หมายถึงสำเร็จ FFH หมายถึงไม่สำเร็จ
การอ่านนาฬิกา RTC	AH = 2CH	CH = ชั่วโมง (0 ถึง 23) CL = นาที (0 ถึง 23) DH = วินาที (0 ถึง 23) DL = 1/100 วินาที (0 ถึง 99)
การตั้งนาฬิกา RTC	AH = 2DH CH = ชั่วโมง (0 ถึง 23) CL = นาที (0 ถึง 23) DH = วินาที (0 ถึง 23) DL = 1/100 วินาที (0 ถึง 99)	AL = 00H หมายถึงสำเร็จ FFH หมายถึงไม่สำเร็จ

ตารางที่ 4
แสดงการเข้าถึงนาฬิกาของ BIOS ผ่านทางอินเทอร์รัพต์ 1AH (Time of Day)

ชื่อฟังก์ชัน	หมายเลขฟังก์ชัน	ค่าที่คืนกลับจากอินเทอร์รัพต์ 1AH
การอ่านนาฬิกา RTC	AH = 02H	CH = ชั่วโมง (BCD) CL = นาที (BCD) DH = วินาที (BCD)
การตั้งนาฬิกา RTC	AH = 03H CH = ชั่วโมง (BCD) CL = นาที (BCD) DH = วินาที (BCD)	ไม่มี
การอ่านปฏิทิน RTC	AH = 04H	CH = ปีศตวรรษ (BCD) CL = ปี ค.ศ. (BCD) DH = เดือน (BCD) DL = วันที่ (BCD)
การตั้งปฏิทิน RTC	AH = 05H CH = ปี ศตวรรษ (BCD) CL = ปี ค.ศ. (BCD) DH = เดือน (BCD) DL = วันที่ (BCD)	ไม่มี

ดังนั้นถ้าหากระบบปฏิบัติการดอสพบว่า ตัวเลขปี ค.ศ. มีค่าน้อยกว่า 1980 ดอสก็จะกำหนดให้เป็นวันที่ 4 มกราคม 1980 โดยอัตโนมัติ และระบบปฏิบัติการดอสจะสนับสนุนปี ค.ศ. ตั้งแต่ 1980 ถึง 2099 เท่านั้น (ทั้งนี้เนื่องจากข้อจำกัดด้านการกำหนดขนาดของหน่วยความจำสำหรับจัดเก็บตัวเลขปี ค.ศ.ของดอส) ดังนั้นโดยทั่วไปในการทดสอบปัญหา Y2K ก็จะพบว่านาฬิกาของระบบปฏิบัติการดอส จะสามารถผ่านปัญหาดังกล่าวไปได้ ดังในรูปที่ 1 โดยทางซ้ายมือจะแสดงนาฬิกาและปฏิทินก่อนปี ค.ศ. 2000 และรูปทางขวามือจะแสดงนาฬิกาและปฏิทินหลังปี ค.ศ. 2000

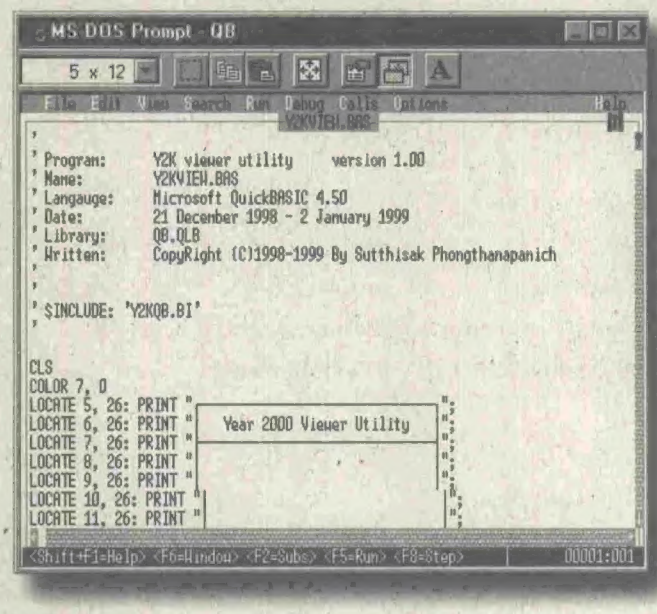
ในการเข้าถึงนาฬิกาของระบบปฏิบัติการดอส สามารถกระทำได้โดยผ่านทางอินเทอร์รัพต์ 21H ของดอสนั้น จะต้องกระทำโดยใช้ฟังก์ชัน 2AH ถึง 2DH ดังที่ปรากฏในตารางที่ 3

ส่วนนาฬิกาของ BIOS นั้นในความเป็นจริงก็เป็นนาฬิกาเรือนเดียวกับ RTC เพราะนาฬิกาของ BIOS เกิดจากการอ่านค่านาฬิกาและปฏิทินจากตำแหน่งข้อมูลใน CMOS RAM ของ RTC โดยผ่านทางอินเทอร์รัพต์ 1AH เท่านั้น ดังนั้นถ้าหาก RTC คำนวณนาฬิกาและปฏิทินผิดพลาด นาฬิกาของ BIOS ก็จะผิดพลาดตามไปด้วย และเท่าที่ผู้เขียนได้ตรวจสอบจากโปรแกรมทดสอบปัญหา Y2K ที่เป็นแชร์แวร์ตามอินเทอร์เน็ตโดยทั่วไป พบว่าโดยส่วนใหญ่จะทดสอบเฉพาะนาฬิกาของระบบปฏิบัติการดอสและของ BIOS เท่านั้น เพราะสามารถเขียนโค้ดโดยผ่านทางการเรียกใช้อินเทอร์รัพต์ได้โดยง่าย แต่ในการเข้าถึงข้อมูลของ RTC จะต้องเขียนโค้ดเพื่อเข้าถึงพอร์ต ซึ่งจะมีความยุ่งยากมากกว่า จึงไม่ค่อยเป็นที่นิยมกัน แต่โปรแกรมในบทความนี้จะสามารถให้ผู้อ่านเข้าถึงนาฬิกาได้ทั้ง 3 ชุดเลยครับ

ในการเข้าถึงนาฬิกาของ BIOS ผ่านทางอินเทอร์รัพต์ 1AH (Time of Day) นั้น จะต้องกระทำโดยใช้ฟังก์ชัน 02H ถึง 05H ดังที่ปรากฏในตารางที่ 4

Advanced BASIC

รูปที่ 1
แสดงผลการทดสอบปัญหา Y2K สำหรับนาฬิกาทั้ง 3 ชุด



รูปที่ 2
แสดงไฟล์ Y2KVIEW.BAS ในเอนไวรอนเมนต์ของ QuickBASIC

รูปที่ 3
แสดงผลลัพธ์ของการรันโปรแกรม Y2KVIEW โดยการไม่กำหนดค่าให้กับ Command Line

โปรแกรม Y2KVIEW.BAS

จากที่กล่าวมาแล้วทั้งหมดข้างต้น เป็นภาคทฤษฎีเกี่ยวกับนาฬิกาและปฏิทินภายใต้ระบบปฏิบัติการดอสของเครื่องพีซีที่ใช้ชิป 80X86 ของอินเทล สำหรับในส่วนสุดท้ายของบทความตอนนี้ ผู้เขียนก็จะอธิบายถึงซอร์สโค้ดของโปรแกรม Y2KVIEW.BAS ที่สามารถใช้ในการตรวจสอบปัญหา Y2K ของเครื่องพีซี โดยถ้าหากผู้อ่านต้องการทดสอบภายใต้เอนไวรอนเมนต์ของ QuickBASIC ผู้อ่านก็ต้องเรียก QuickBASIC โดยกำหนดชื่อของไลบรารีเป็น Y2KQB.QLB ดังนี้

```
C:\>qb /l y2kqb.qlb
```

เนื่องจากฟังก์ชันทั้งหมดที่จำเป็นในการทดสอบปัญหา Y2K ของเครื่องพีซีได้ถูกเขียนเป็นโพรซีเดอร์เก็บเอาไว้ในไลบรารีเป็น Y2KQB.QLB จนครบหมดแล้ว (ไลบรารีเป็น Y2KQB.QLB เป็นไฟล์ที่สร้างจากไฟล์ Y2KQB.OBJ และ QB.LIB) จากนั้นก็ให้ผู้อ่านเรียกไฟล์ Y2KVIEW.BAS ขึ้นมาดังในรูปที่ 2 ซึ่งจะเห็นว่าเป็นโปรแกรมง่าย ๆ ที่ผู้อ่านจะสามารถทำความเข้าใจได้อย่างรวดเร็ว โดยไม่จำเป็นต้องยุ่งเกี่ยวกับการเรียกอินเทอร์รัพต์แต่อย่างใด เพราะในส่วนที่เกี่ยวกับการเรียกอินเทอร์รัพต์ และการแปลงค่า BCD นั้น ผู้เขียนได้รวบรวมเอาไว้ในไฟล์ Y2KQB.OBJ จนหมดแล้ว

ในการรันโปรแกรม Y2KVIEW ผู้อ่านสามารถกระทำได้ 2 วิธี โดยการไม่ต้องกำหนดค่าให้กับ Command Line และโดยการกำหนดค่า "/Y2K" ให้กับ Command Line (การกำหนดค่าให้กับ Command Line สามารถกระทำได้โดยผ่านทางคำสั่ง Modify COMMAND$ ของเมนู Run) ซึ่งวิธีทั้งสองมีความหมายดังนี้ วิธีแรกเป็นการกำหนดให้โปรแกรม Y2KVIEW แสดงผลนาฬิกาและปฏิทินปัจจุบันของทั้ง 3 ชุด (RTC, BIOS และ DOS) ดังในรูปที่ 3 เป็นตัวอย่างผลลัพธ์ที่ได้ ส่วนวิธีหลังเป็นการกำหนดให้โปรแกรม Y2KVIEW แสดงผลนาฬิกาและปฏิทินของวันที่ 31 มกราคม 1999 เวลา 23:59:55 เพื่อทดสอบปัญหา Y2K นั่นเอง (ดูตัวอย่างผลลัพธ์ได้จากรูปที่ 1)

ส่วนโพรซีเดอร์ทั้งหมดที่มีในไฟล์ Y2KQB.OBJ ได้ถูกประกาศเอาไว้ในไฟล์ Y2KQB.BI โดยมีรายละเอียดดังต่อไปนี้

Advanced BASIC

GetBIOSDate$

การประกาศ	DECLARE FUNCTION GetBIOSDate$ ()
หน้าที่	อ่านปฏิทินวันที่จากนาฬิกา BIOS
พารามิเตอร์	ไม่มี
การเรียกใช้	BIOSDate$ = GetBIOSDate$()
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา BIOS ในรูปแบบ mm-dd-yyyy
หมายเหตุ	ได้จากการเรียกใช้อินเทอร์รัพต์ 1AH หมายเลขฟังก์ชัน 04H

GetBIOSTime$

การประกาศ	DECLARE FUNCTION GetBIOSTime$ ()
หน้าที่	อ่านเวลาจากนาฬิกา BIOS
พารามิเตอร์	ไม่มี
การเรียกใช้	BIOSTime$ = GetBIOSTime$()
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา BIOS ในรูปแบบ hh-mm-ss
หมายเหตุ	ได้จากการเรียกใช้อินเทอร์รัพต์ 1AH หมายเลขฟังก์ชัน 02H

GetDOSDate$

การประกาศ	DECLARE FUNCTION GetDOSDate$ ()
หน้าที่	อ่านปฏิทินวันที่จากนาฬิกา DOS
พารามิเตอร์	ไม่มี
การเรียกใช้	DOSDate$ = GetDOSDate$()
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา DOS ในรูปแบบ mm-dd-yyyy
หมายเหตุ	ได้จากการเรียกใช้อินเทอร์รัพต์ 21H หมายเลขฟังก์ชัน 2AH

GetDOSTime$

การประกาศ	DECLARE FUNCTION GetDOSTime$ ()
หน้าที่	อ่านเวลาจากนาฬิกา DOS
พารามิเตอร์	ไม่มี
การเรียกใช้	DOSTime$ = GetDOSTime$()
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา DOS ในรูปแบบ hh-mm-ss
หมายเหตุ	ได้จากการเรียกใช้อินเทอร์รัพต์ 21H หมายเลขฟังก์ชัน 2CH

GetRTCDate$

การประกาศ	DECLARE FUNCTION GetRTCDate$ ()
หน้าที่	อ่านปฏิทินวันที่จากนาฬิกา RTC
พารามิเตอร์	ไม่มี
การเรียกใช้	RTCDate$ = GetRTCDate$()
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา RTC ในรูปแบบ mm-dd-yyyy
หมายเหตุ	ได้จากการเรียกผ่านพอร์ต 70H และ 71H

Advanced BASIC

GetRTCTime$

การประกาศ	`DECLARE FUNCTION GetRTCTime$ ()`
หน้าที่	อ่านเวลาจากนาฬิกา RTC
พารามิเตอร์	ไม่มี
การเรียกใช้	`RTCTime$ = GetRTCTime$()`
การคืนกลับค่า	สตริง ปฏิทินวันที่จากนาฬิกา RTC ในรูปแบบ hh-mm-ss
หมายเหตุ	ได้จากการเรียกผ่านพอร์ต 70H และ 71H

PhraseCOMMAND

การประกาศ	`DECLARE SUB PhraseCOMMAND (cmdline$)`
หน้าที่	อ่าน command lind "/Y2K" และกำหนดนาฬิกาและปฏิทิน เพื่อทดสอบปัญหา Y2K
พารามิเตอร์	ไม่มี
การเรียกใช้	`cmdline$ = COMMAND$` `PhraseCOMMAND cmdline$`
การคืนกลับค่า	ไม่มี
หมายเหตุ	ผู้อ่านสามารถเขียนโค้ดเพื่อทดสอบปัญหา Y2K โดยการกำหนดนาฬิกาและปฏิทินให้กับฟังก์ชัน DATE$ และ TIME$ ของ QuickBASIC ดังนี้ DATE$ = "12-31-1999" TIME$ = "23:59:55"

RestoreCurrentDateTime

การประกาศ	`DECLARE SUB RestoreCurrentDateTime (sDate$, sTime$, Hr%, Mn%, Sc%)`	
หน้าที่	คืนกลับนาฬิกาและปฏิทิน RTC เป็นค่าของก่อนที่จะถูกแก้ไขเพื่อทดสอบปัญหา Y2K	
พารามิเตอร์	sDate$	หมายถึง ตัวแปรสำหรับจัดเก็บค่าปฏิทินวันที่
	sTime$	หมายถึง ตัวแปรสำหรับจัดเก็บค่านาฬิกาเวลา
	Hr%, Mn%, Sc%	หมายถึง ตัวแปรสำหรับจัดเก็บค่านาฬิกาเวลา (ชั่วโมง นาที และวินาที ตามลำดับ)
การเรียกใช้	`RestoreCurrentDateTime sDate$, sTime$, Hr%, Mn%, Sc%`	
การคืนกลับค่า	ไม่มี	
หมายเหตุ	ค่าตัวแปรที่กำหนดให้กับโพรซีเดอร์นี้ ควรจะต้องถูกอ่านจากโพรซีเดอร์ SaveCurrentDateTime เท่านั้น	

SaveCurrentDateTime

การประกาศ	`DECLARE SUB SaveCurrentDateTime (sDate$, sTime$, Hr%, Mn%, Sc%)`	
หน้าที่	จัดเก็บนาฬิกาและปฏิทิน RTC ก่อนที่จะถูกแก้ไขเพื่อทดสอบปัญหา Y2K	
พารามิเตอร์	sDate$	หมายถึง ตัวแปรสำหรับจัดเก็บค่าปฏิทินวันที่ปัจจุบัน
	sTime$	หมายถึง ตัวแปรสำหรับจัดเก็บค่านาฬิกาเวลาปัจจุบัน
	Hr%, Mn%, Sc%	หมายถึง ตัวแปรสำหรับจัดเก็บค่านาฬิกาเวลาปัจจุบัน (ชั่วโมง นาที และวินาที ตามลำดับ)
การเรียกใช้	`Dim sDate$, sTime$, Hr%, Mn%, Sc%` `SaveCurrentDateTime sDate$, sTime$, Hr%, Mn%, Sc%`	
การคืนกลับค่า	ผ่านทางตัวแปรทั้งหมดที่ส่งให้กับโพรซีเดอร์	
หมายเหตุ	ไม่มี	

Advanced BASIC

UpdateTime	
การประกาศ	DECLARE SUB UpdateTime (Hr%, Mn%, Sc%)
หน้าที่	ปรับปรุงค่านาฬิกาและปฏิทินปัจจุบันให้ทันสมัย ในขณะกำลังทดสอบปัญหา Y2K
พารามิเตอร์	Hr%, Mn%, Sc% หมายถึง ตัวแปรสำหรับจัดเก็บค่านาฬิกาเวลาปัจจุบัน (ชั่วโมง นาทีและวินาที ตามลำดับ)
การเรียกใช้	UpdateTime Hr%, Mn%, Sc%
การคืนกลับค่า	ผ่านทางตัวแปรทั้งหมดที่ส่งให้กับโพรซีเดอร์
หมายเหตุ	ให้เรียกใช้โพรซีเดอร์นี้ เฉพาะภายในส่วนที่ถูกเรียกโดยเหตุการณ์ Timer(1) เท่านั้น

และเนื่องจากโปรแกรม Y2KVIEW เป็นโปรแกรมง่าย ๆ ที่เรียกใช้โพรซีเดอร์ต่าง ๆ จากไฟล์ Y2KQB.OBJ ไปใช้งาน ดังนั้นเพื่อเป็นการประหยัดกระดาษจึงไม่ขออธิบายรายละเอียดของซอร์สโค้ดในบทความนี้ แต่ขอให้ผู้อ่านศึกษาเองจากลิสติงสั้น ๆ ส่วนในการคอมไพล์เพื่อสร้างไฟล์ Y2KVIEW.EXE ก็สามารถกระทำได้โดยใช้ BC.EXE และ LINK.EXE โดยผู้อ่านสามารถที่จะศึกษาวิธีใช้งานได้จากหนังสือ "การใช้โปรแกรม QuickBASIC" ซึ่งแต่งโดยผู้เขียนเช่นกัน และถ้าหากผู้อ่านท่านใดสงสัยหรือพบข้อบกพร่องของโปรแกรม Y2KVIEW หรือสามารถพัฒนาให้โปรแกรม Y2KVIEW มีความสามารถมากขึ้น ก็ช่วยเขียนเมล์มาคุยกันบ้าง ที่แอดเดรสแอดเดรสอี-เมล์ sutthisakp@yahoo.com เพื่อที่จะได้เป็นการช่วยกันกระจายความรู้เป็นวิทยาทานให้กับสังคมครับ พบกันใหม่ในตอนหน้า สวัสดีครับ

```
'
' Program:     Year2000 viewer utility include file   version 1.00
' Name:        Y2KQB.BI
' Langauge:    Microsoft QuickBASIC 4.50
' Date:        21 December 1998 - 2 January 1999
' Library:     Y2KQB.QLB
' Written:     CopyRight (C)1998-1999 By Sutthisak Phongthanapanich
'
'
' $INCLUDE: 'QB.BI'
'

DECLARE FUNCTION GetBIOSDate$ ()
DECLARE FUNCTION GetBIOSTime$ ()
DECLARE FUNCTION GetDOSDate$ ()
DECLARE FUNCTION GetDOSTime$ ()
DECLARE FUNCTION GetRTCDate$ ()
DECLARE FUNCTION GetRTCTime$ ()

DECLARE SUB PhraseCOMMAND (cmdline$)
DECLARE SUB RestoreCurrentDateTime (sDate$, sTime$, Hr%, Mn%, Sc%)
DECLARE SUB SaveCurrentDateTime (sDate$, sTime$, Hr%, Mn%, Sc%)
DECLARE SUB UpdateTime (Hr%, Mn%, Sc%)
```

ลิสติงที่ 1 ไฟล์อินคลูด Y2KQB.BI

ลิสติงที่ 2
ไฟล์โปรแกรม Y2KVIEW.BAS

```
'
' Program:     Y2K viewer utility     version 1.00
' Name:        Y2KVIEW.BAS
' Langauge:    Microsoft QuickBASIC 4.50
' Date:        21 December 1998 - 2 January 1999
' Library:     QB.QLB
' Written:     CopyRight (C)1998-1999 By Sutthisak Phongthanapanich
'
'
' $INCLUDE: 'Y2KQB.BI'
'

CLS
COLOR 7, 0
LOCATE 5, 26: PRINT "??????????????????????????????????";
LOCATE 6, 26: PRINT "?  Year 2000 Viewer Utility  ?";
LOCATE 7, 26: PRINT "??????????????????????????????????";
LOCATE 8, 26: PRINT "?                            ?";
LOCATE 9, 26: PRINT "?                            ?";
LOCATE 10, 26: PRINT "?                           ?";
LOCATE 11, 26: PRINT "?                           ?";
LOCATE 12, 26: PRINT "?                           ?";
LOCATE 13, 26: PRINT "??????????????????????????????????";
LOCATE 14, 26: PRINT "?  CopyRight (C)1998-1999   ?";
LOCATE 15, 26: PRINT "?  Sutthisak Phongthanapanich  ?";
LOCATE 16, 26: PRINT "??????????????????????????????????";
COLOR 15, 9
LOCATE 6, 27: PRINT "  Year 2000 Viewer Utility  ";
COLOR 15, 6
LOCATE 14, 27: PRINT "  CopyRight (C)1998-1999  ";
LOCATE 15, 27: PRINT "  Sutthisak Phongthanapanich  ";

ON ERROR GOTO ErrTrap
```

Advanced BASIC

```
' Save current Date and Time.
SaveCurrentDateTime sDate$, sTime$, Hr%, Mn%, Sc%
' Check command line /Y2K.
PhraseCOMMAND COMMAND$

COLOR 15 + 16, 0
LOCATE 18, 26: PRINT "<< Press any key to terminate >>";
COLOR 7, 0

' Update Timer each 1 second.
TIMER ON
ON TIMER(1) GOSUB UpdateTimer

' Wait for any key pressed.
DO
LOOP WHILE INKEY$ = ""

' Restore current Date and Time if /Y2K command line specified.
IF LEN(COMMAND$) THEN
   LOCATE 20, 1: PRINT "Restore date and time please wait...";
   RestoreCurrentDateTime sDate$, sTime$, Hr%, Mn%, Sc%
   LOCATE 21, 1: PRINT "Date and time restored.";
END IF

COLOR 7, 0
LOCATE 22, 1
END

UpdateTimer:
   ' Update date and time in clock display box.
   COLOR 12, 0
   LOCATE 9, 28: PRINT "? DOS  ? ";
   COLOR 15, 0: PRINT GetDOSDate$; " "; GetDOSTime$
   COLOR 12, 0
   LOCATE 10, 28: PRINT "? BIOS ? ";
   COLOR 15, 0: PRINT GetBIOSDate$; " "; GetBIOSTime$
   COLOR 12, 0
   LOCATE 11, 28: PRINT "? RTC  ? ";
   COLOR 15, 0: PRINT GetRTCDate$; " "; GetRTCTime$

   ' Update time for restore when program terminate.
   UpdateTime Hr%, Mn%, Sc%
   RETURN

ErrTrap:
   ' Trap error in any cases, esp. command line incorrect.
   BEEP
   COLOR 14 + 16, 0
   LOCATE 19, 1: PRINT "? Error occured in program."
   PRINT "  Using Example : C:\>RTC2000 "
   PRINT "                  or : C:\>RTC2000 /Y2K"
   COLOR 7, 0
   PRINT
   END
```

เทคนิคประจำคอลัมน์

ถาม ขอทราบผลที่ได้จากโปรแกรม Y2KVIEW ในการทดสอบปัญหา Y2K ว่าก่อนบูตเครื่องกับหลังบูตเครื่อง นาฬิกาและปฏิทินจะตรงกันหรือไม่?

ตอบ นาฬิกาและปฏิทินของ RTC และ BIOS จะตรงกันเสมอ ไม่ว่าก่อนบูตเครื่องกับหลังบูตเครื่อง และถ้าหากไม่ผ่านปัญหา Y2K ปี ค.ศ. ที่ได้ก็จะเท่ากับ 1900

แต่สำหรับนาฬิกาและปฏิทินของระบบปฏิบัติการดอสจะไม่ตรงสำหรับก่อนบูตเครื่องกับหลังบูตเครื่อง และถ้าหากไม่ผ่านปัญหา Y2K ก่อนบูตเครื่องปี ค.ศ. ที่ได้ก็จะเท่ากับ 2000 แต่หลังบูตเครื่องปี ค.ศ. ที่ได้ก็จะเท่ากับ 1980 เนื่องจากถ้าหากไม่ผ่านปัญหา Y2K หลังจากบูตเครื่องแล้ว RTC จะกำหนดให้ปี ค.ศ. ที่ได้ก็จะเท่ากับ 1900 ดังนั้นเนื่องจากปี ค.ศ. น้อยกว่า 1980 ดังนั้นดอสจึงกำหนดให้นาฬิกาและปฏิทินเป็นวันที่ 4 มกราคม 1980 แทน ยกเว้นเครื่องที่ BIOS ได้รับการแก้ไขบ้างแล้ว เช่น BIOS ของ Award Software International Inc. เป็นต้น โดยเครื่องของผู้เขียนหลังจากบูตเครื่องแล้ว BIOS จะกำหนดให้ RTC เป็นปี ค.ศ. เท่ากับ 1994 แทนที่จะใช้ค่า 1900 ของ RTC เป็นต้น